Making a Career in Malware Analysis & Reverse Engineering

Tear apart malicious code. Understand how attackers really work.

AUTHOR	Babashaheer
VERSION	1.0
DATE	April 2026
SERIES	Cybersecurity Career Series — Document 04 of 09
AUDIENCE	Students and security professionals interested in malware and reverse engineering

Contents

Baba Shaheer Gutappa
Digitally signed by Baba Shaheer Gutappa
Date: 2026.04.18 11:55:08 +01'00'

1. What Is Malware Analysis & Reverse Engineering?

Every piece of malicious software that causes damage to a business, steals data or holds a hospital to ransom was written by someone. Malware analysts are the people who take that software apart — piece by piece — to understand exactly how it works, what it does, and how to detect and stop it.

Reverse engineering is the process of taking a compiled binary (a program with no readable source code) and working backwards to understand its logic. This is one of the most technically demanding disciplines in all of cybersecurity. It requires patience, strong programming knowledge, comfort reading low-level machine code, and a genuine curiosity about how software works at its deepest level.

> **Malware analysts are the people who open the dangerous thing and study it.**
> They work with ransomware, spyware, rootkits and banking trojans that have caused millions of pounds of damage — safely, in isolated lab environments.

The two disciplines and how they relate

	Malware Analysis	Reverse Engineering
Focus	Understanding what malware does — its behaviour, targets and impact	Understanding how software works at the code level — structure, logic, algorithms
Output	Threat report, IOCs, YARA rules, detection signatures	Decompiled code, logic diagrams, understanding of algorithms
Key tools	Sandbox environments, network monitors, process monitors	Ghidra, IDA Pro, x64dbg, Binary Ninja
Overlap	All malware analysis uses RE techniques — they are inseparable in practice	RE is used for malware, but also for vulnerability research, CTF and software cracking
Typical employer	Antivirus vendors, threat intelligence firms, government agencies	The same — plus specialist security consulting firms and research teams

Table 1: Malware analysis and reverse engineering — two overlapping disciplines

Level	Typical UK Salary	Roles
Junior	£32,000 – £48,000	Junior Malware Analyst, Threat Intelligence Analyst, SOC Analyst (Malware tier)
Mid-level	£50,000 – £75,000	Malware Analyst, Reverse Engineer, Threat Researcher
Senior	£75,000 – £100,000+	Senior Malware Researcher, Principal Reverse Engineer, Threat Intelligence Lead
Specialist	£80,000 – £120,000+	Vulnerability Researcher, State-level threat analyst, Exploit developer (defensive)

Table 2: UK salary ranges for malware analysis and reverse engineering roles (CW Jobs, 2024)

2. Types of Malware You Will Encounter

Before analysing malware, you need to understand the landscape. Malware is not one thing — it is a broad category of malicious software with different objectives, behaviours and technical implementations. A ransomware analyst's day looks very different from a rootkit researcher's.

Type	What It Does	Notable Examples	Technical Complexity
Ransomware	Encrypts victim files and demands payment for the decryption key. May also exfiltrate data first (double extortion).	WannaCry, LockBit, Conti, REvil	Medium — well-documented technique
Banking Trojan	Steals banking credentials using web injection, keylogging or form-grabbing. Often modular and stealthy.	Emotet, TrickBot, Dridex, Zeus	High — evasion and injection heavy
RAT (Remote Access Trojan)	Gives an attacker full remote control of a victim machine — file access, camera, keylogger, screenshot.	njRAT, AsyncRAT, Gh0st RAT	Medium — network comms focused
Rootkit	Hides itself and other malware from the OS and security tools. Operates at kernel level.	TDL4, Necurs, BlackEnergy	Very High — kernel-mode expertise needed
Worm	Self-replicating malware that spreads across networks without user interaction. Often carries a payload.	WannaCry (worm component), Conficker	Medium — network propagation logic
Spyware / Infostealer	Silently collects sensitive information — passwords, clipboard content, browser data.	RedLine Stealer, Raccoon, Vidar	Medium — data exfiltration focused
Bootkit	Infects the Master Boot Record or UEFI firmware. Survives OS reinstallation.	BlackLotus (UEFI bootkit)	Very High — pre-OS expertise needed
Fileless Malware	Operates entirely in memory — no file dropped on disk. Lives in PowerShell, WMI or injected process.	PowerSploit, Cobalt Strike (fileless mode)	High — memory and evasion focused

Table 3: Major malware families and their technical characteristics

3. Static Analysis — Without Running the Code

Static analysis examines a malware sample without executing it. You look at the file itself — its structure, strings, imported functions, embedded resources and code — to understand what it might do. This is always the first step. You never run an unknown sample without understanding it first, and even then only in a fully isolated sandbox environment.

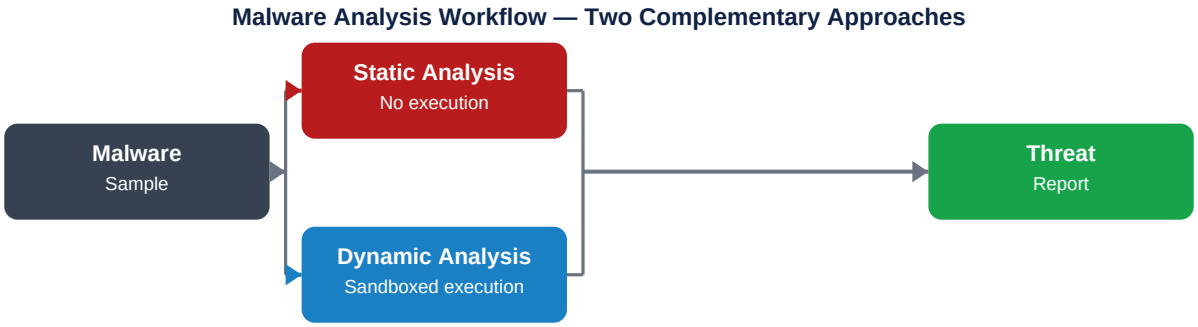


Figure 1: Malware analysis workflow. Static and dynamic analysis are complementary — neither alone is sufficient.

Step-by-step static analysis process

■ **1. Hash the sample.** Before touching the file, generate its MD5, SHA1 and SHA256 hashes. Search these on VirusTotal, MalwareBazaar and Hybrid Analysis. If it is a known sample, you already have threat intelligence before you have done any analysis.

■ **2. Check the file type.** The extension tells you nothing — attackers routinely mislabel files. Use the 'file' command (Linux) or ExeinfoPE (Windows) to check the true file format. Is it a PE (Windows executable), ELF (Linux), script, document or something else?

■ **3. Check packing and obfuscation.** Most real-world malware is packed or obfuscated to resist analysis. Tools: PEiD, Detect-It-Easy (DiE), ExeinfoPE. If it is packed, you need to unpack it before meaningful static analysis is possible.

■ **4. Extract strings.** The 'strings' command extracts all printable sequences from the binary. Look for URLs, IP addresses, domain names, registry keys, file paths, error messages and hardcoded credentials. Even heavily obfuscated malware often has readable strings.

■ **5. Examine PE headers and imports.** The Import Address Table (IAT) lists every Windows API function the binary calls. Functions like CreateRemoteThread, VirtualAllocEx, WriteProcessMemory indicate process injection. InternetOpenA, HttpSendRequest indicate network communication.

■ **6. Disassemble and decompile.** Load the unpacked binary into Ghidra or IDA Pro. The disassembler converts machine code to assembly. The decompiler generates readable C-like pseudocode. This is where the real analysis begins — reading the code to understand the logic.

What to look for in the Import Address Table (IAT)

API Function	DLL	What It Signals
CreateRemoteThread	kernel32.dll	Process injection — running code inside another process

API Function	DLL	What It Signals
VirtualAllocEx + WriteProcessMemory	kernel32.dll	Classic shellcode injection pattern
RegSetValueEx / RegCreateKeyEx	advapi32.dll	Persistence — writing to Windows registry for autorun
InternetOpenA / WinHttpConnect	wininet.dll / winhttp.dll	Network communication — C2 beacon or data exfiltration
CryptEncrypt / CryptGenKey	advapi32.dll / bcrypt.dll	Encryption — potentially ransomware or comms encryption
SetWindowsHookEx	user32.dll	Keylogging — intercepting keyboard input
NtQuerySystemInformation	ntdll.dll	Anti-analysis / rootkit behaviour — querying system state
IsDebuggerPresent / CheckRemoteDebuggerPresent	kernel32.dll	Anti-debugging — evading analysis tools

Table 4: Windows API functions that indicate malicious behaviour

4. Dynamic Analysis — Watching It Behave

Dynamic analysis executes the malware in a controlled, isolated environment — a sandbox — and monitors everything it does. File operations, registry changes, network connections, process creation, API calls. You learn what it actually does at runtime, which is often different from what static analysis suggests (especially with obfuscated or packed samples that decrypt their real payload in memory).

CRITICAL: Never run unknown malware on a production system or your main machine.
Always use an isolated VM with no network connectivity to your main environment.
Snapshot before running. Revert to snapshot after analysis. Never keep a live infected VM.

What to Monitor	Tool Used	What You Are Looking For
Process activity	Process Monitor (Procmon), Process Hacker	New processes spawned, process injection, unusual parent-child relationships
File system changes	Procmon, Noriben	Files created, deleted or modified. New executables dropped. Ransom notes written.
Registry changes	Procmon, RegShot	Autorun keys written (HKCU\Run, HKLM\Run). New services registered. Configuration stored.
Network traffic	Wireshark, INetSim, Fakenet-NG	DNS queries to suspicious domains. HTTP/S POST to C2 server. Data exfiltration patterns.
API calls	API Monitor, x64dbg with logging	Which Windows API functions are called and with what parameters — more detail than static IAT
Memory	Volatility on memory dump taken during execution	Injected code in other processes. Decrypted payload. C2 configuration stored in memory.

Table 5: Dynamic analysis monitoring categories, tools and what to look for

Automated sandboxes for initial triage

- **ANY.RUN (any.run):** Interactive online sandbox. Watch malware execute in real time. Excellent for quick initial triage of suspicious files or URLs. Free tier available.
- **Hybrid Analysis (hybrid-analysis.com):** Automated sandbox using Falcon Sandbox engine. Detailed behavioural report, YARA matches, network indicators. Free.
- **MalwareBazaar (bazaar.abuse.ch):** Database of malware samples with community analysis. Good for finding existing analysis of known samples.
- **Cuckoo Sandbox:** Open-source automated sandbox you can deploy locally. Full control over the environment. Recommended for sensitive samples you cannot submit publicly.

5. The Assembly Language You Actually Need

You do not need to write assembly. You need to read it well enough to follow what the disassembler shows you. The good news: most real-world malware analysis requires a relatively limited subset of x86/x64 assembly. Master these concepts and you can follow the vast majority of malware logic.

Essential registers and what they do

Register (x64)	32-bit	What It Stores / Is Used For
RAX	EAX	Accumulator — stores function return values. Check RAX after a call to see what was returned.
RCX	ECX	First function argument (Windows x64 calling convention). Counter in loops.
RDX	EDX	Second function argument. Also used in division.
RSP	ESP	Stack Pointer — always points to the top of the current stack frame.
RBP	EBP	Base Pointer — marks the base of the current stack frame. Used to access local variables.
RIP	EIP	Instruction Pointer — the address of the next instruction to execute. Manipulated in exploits.
RBX, RSI, RDI	EBX, ESI, EDI	General purpose. Used for loop counters, source/destination in memory operations.

Table 6: Key x86/x64 registers for malware analysis

Instructions you will see constantly

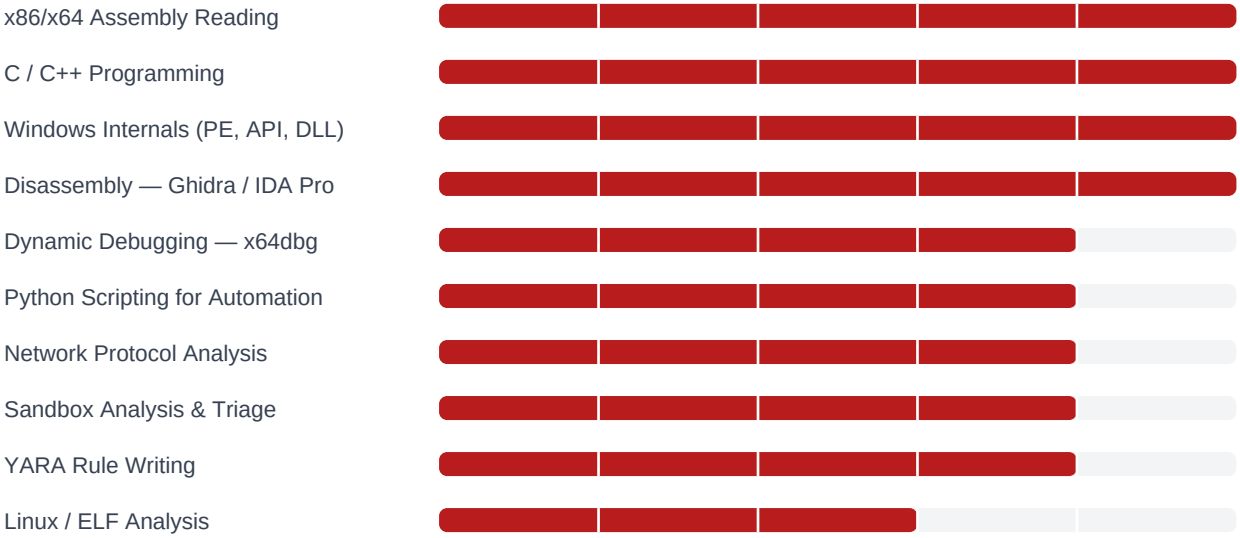
malware_analysis.py

```
; Function prologue — sets up stack frame
push    rbp
mov     rbp, rsp
sub     rsp, 0x40        ; Allocate 64 bytes of local space

; Check if debugger is present (anti-analysis)
call    IsDebuggerPresent
test    eax, eax         ; If eax == 0, no debugger
jnz     exit_malware     ; If debugger found, exit quietly

; Loop: encrypt each byte of a buffer
xor     ecx, ecx         ; ecx = 0 (loop counter)
encrypt_loop:
  xor   byte [rbx+rcx], 0x42  ; XOR encrypt with key 0x42
  inc   ecx
  cmp   ecx, 0x100       ; Loop 256 times
```

Code 1: Annotated assembly excerpt showing common patterns — stack setup, anti-debug check, XOR encryption loop

6. Core Skills and Tools

Skills expected at mid-level malware analyst:



Tool	Category	What It Does	Cost
Ghidra	Disassembler / Decompiler	NSA-developed open-source reverse engineering tool. Disassembles and decompiles to C-like pseudocode. Feature-comparable to IDA Pro.	Free
IDA Pro	Disassembler / Decompiler	The industry standard for professional reverse engineering. Powerful scripting, plugin ecosystem, excellent for complex binaries.	Commercial (IDA Free available)
x64dbg	Debugger	Open-source debugger for Windows x64/x32 binaries. Use alongside Ghidra for dynamic tracing of execution flow.	Free
Binary Ninja	Disassembler	Modern RE platform with excellent API for automation. Good alternative to IDA Pro.	Commercial (free cloud version)
Detect-It-Easy (DiE)	Packer Detection	Identifies packers, protectors and compilers used. Essential first step before static analysis.	Free
PE Bear / PE Studio	PE Analysis	Inspect PE file headers, sections, imports, exports, overlay data and resources without executing.	Free
Wireshark + INetSim	Network Analysis	Wireshark captures traffic. INetSim simulates internet services so malware's network calls are captured safely.	Free

Tool	Category	What It Does	Cost
Process Monitor (Procmon)	Behaviour Monitoring	Real-time monitoring of file system, registry and process/thread activity during dynamic analysis.	Free (Sysinternals)
Cuckoo Sandbox	Automated Sandbox	Open-source automated malware analysis — deploys a VM, runs the sample, returns a full behaviour report.	Free
FLOSS	String Extraction	FireEye FLOSS extracts obfuscated strings that the standard 'strings' command misses — including decoded/decrypted content.	Free
YARA	Detection Rules	Write signature-based rules to detect malware families based on strings, byte patterns or structural features.	Free

Table 7: Core tools for malware analysis and reverse engineering

7. Writing YARA Rules for Detection

YARA is the malware analyst's signature language. A YARA rule describes patterns that identify a specific malware family — byte sequences, strings, file structure features. Once written, a YARA rule can be used to scan files, memory dumps and network traffic for malware matching that pattern. Understanding how to write and read YARA rules is one of the most practical skills you can develop.

malware_analysis.py

```
rule BankingTrojan_Dridex_Variant {
    meta:
        description = "Detects Dridex loader based on PE structure + strings"
        author      = "Babashaheer"
        date        = "2026-04-10"
        reference   = "https://malpedia.caad.fkie.fraunhofer.de/details/win.dridex"

    strings:
        $str1 = "Bot ID:" wide ascii
        $str2 = "Loader32" wide ascii
        $hex1 = { 55 8B EC 83 C4 F8 53 56 57 }  // Prologue pattern
        $hex2 = { E8 ?? ?? ?? ?? 85 C0 74 }      // Conditional jump pattern

    condition:
        uint16(0) == 0x5A4D        // MZ header (PE file)
        and filesize < 500KB
        and (1 of ($str*))         // At least one string match
        and (1 of ($hex*))         // At least one hex pattern
```

Code 2: Example YARA rule structure for detecting a banking trojan variant

Good YARA rules are specific enough to avoid false positives.
Test every rule against a corpus of clean files before deploying it in production.
The YARA documentation and VirusTotal Intelligence YARA rules are the best references.

8. Career Paths in Malware Analysis

Malware analysis is one of the more specialised career paths in cybersecurity. It is not a typical entry-level role — most analysts come in with at least 2–3 years of SOC, incident response or security research experience. But the progression, once started, is rapid.

Role	What You Do	Where You Work	UK Salary
Tier 2/3 SOC Analyst (Malware tier)	Escalated alert triage involving malware. Basic static analysis. Signature writing.	MSSP, large enterprise SOC	£35k–£50k
Malware Analyst	Full static and dynamic analysis of samples. Threat reports. IOC extraction. YARA rules.	Threat intelligence firms, AV vendors	£48k–£70k
Threat Intelligence Analyst	Contextualise malware findings into threat intelligence. Track threat actor TTPs. Report to security teams.	Threat intel firms, banks, government	£50k–£75k
Reverse Engineer	Deep binary analysis. Unpacking, deobfuscation, reconstructing malware logic. Protocol reverse engineering.	Specialist security firms, GCHQ, defence	£60k–£90k
Vulnerability Researcher	Find new vulnerabilities in software. Sometimes overlaps with malware analysis when studying exploits.	Security vendors, bug bounty, government	£70k–£100k+
Principal Malware Researcher	Lead research into advanced malware families. Publish technical reports. Conference speaker.	CrowdStrike, Mandiant, NCSC, academia	£85k–£120k+

Table 8: Career paths in malware analysis

9. The Learning Roadmap

Malware analysis has the steepest learning curve of any cybersecurity career path. There are no shortcuts — you need strong foundations in programming, operating systems and networking before you can meaningfully read disassembled code. Be patient with the process. Every analyst started unable to read assembly.

1 — Build strong C / Python programming skills
You must be able to read C code fluently and write Python scripts to automate tasks. If you cannot read a C function and understand what it does, disassembled code will be meaningless. Work through a C programming course. 6–8 weeks.

2 — Learn Windows internals
Understand PE file format, the Windows API, DLLs, the registry, process memory layout, and how Windows loads executables. The 'Windows Internals' book (Russinovich) is the definitive reference. Online: OS Internals courses on Pluralsight. 4–6 weeks.

3 — Learn x86/x64 assembly basics
You do not need to write assembly — you need to read it. Start with OpenSecurityTraining2 (free, excellent). Focus on registers, stack operations, calling conventions, and common patterns you will see in compiled code. 4–6 weeks.

4 — Install Ghidra and start reversing
Download Ghidra (free from NSA). Find beginner crackme challenges on crackmes.one. These are intentionally reversible programs designed for practice — no malware involved. Work through at least 20 crackmes before touching real malware. Ongoing.

5 — Start with basic malware triage
Download samples from MalwareBazaar (safe — these are known samples with community analysis). Run them through ANY.RUN sandbox. Read the reports. Compare to your own static analysis. Start recognising patterns. Ongoing.

6 — Build your malware lab
Create an isolated VM environment with Windows (target), REMnux (Linux analysis OS, free), and INetSim for fake internet services. Snapshot before every experiment. Flare-VM (free) sets up a complete Windows RE environment automatically. 1–2 weeks.

7 — Work through malware analysis courses
TCM Security's Practical Malware Analysis & Triage (PMAT) is the best affordable practical course available. ANY.RUN Academy is free. Malware Unicorn workshops are free and excellent. 6–8 weeks.

8 — Write reports and apply
Analyse a real malware sample from MalwareBazaar. Write a full professional report — static analysis, dynamic analysis, IOCs, YARA rule, recommended detections. This becomes your portfolio. Apply for Tier 2/3 SOC and malware analyst roles.

Realistic timeline from zero to junior malware analyst: 18–24 months.
From security background (SOC/IR) with some RE exposure: 9–14 months.
This is a long path — but the scarcity of qualified analysts means the payoff is significant.

10. Certifications That Matter

Malware analysis certifications are rarer and more practical than those in other cybersecurity disciplines. Employers in this field care far more about demonstrated skill — reports you have written, samples you have analysed, tools you can use — than about paper certifications. That said, the following are respected:

Certification	Level	Provider	Focus	Respect Level
GREM — GIAC Reverse Engineering Malware	Mid–Senior	GIAC / SANS	Static and dynamic analysis, anti-analysis techniques, memory forensics	Very High — the gold standard for malware analysts
PMAT — Practical Malware Analysis & Triage	Beginner–Mid	TCM Security	Practical hands-on malware triage — static, dynamic, YARA, report writing	Growing fast — highly practical and affordable
CompTIA Security+	Beginner	CompTIA	Broad security foundations — not specific to malware but often required as a baseline	Medium — prerequisite for many job applications
GCFE / GCFA — GIAC Forensics	Mid	GIAC / SANS	Digital forensics — complementary to malware analysis, especially for incident response	High — good complement to GREM
CHFI — Computer Hacking Forensic Investigator	Mid	EC-Council	Digital forensics covering malware analysis components	Medium-High — good breadth
eWPT / eCMAP — eLearnSecurity	Mid	INE / eLearnSecurity	eCMAP specifically covers malware analysis — practical exam format	Growing — practical certification with real-world focus

Table 9: Certifications for malware analysts in order of progression

Portfolio over certifications in this field.
A published malware analysis report carries more weight than a certificate.
Contribute write-ups to ANY.RUN, MalwareBazaar, or publish on your own blog.

11. Case Study — Analysing a Banking Trojan

> This is a fictional case study based on real banking trojan analysis techniques.
> The malware behaviour described mirrors real-world Dridex/TrickBot-family characteristics.

The Sample

A malware analyst at a UK bank's threat intelligence team receives a suspicious **.docm** (macro-enabled Word document) that was caught by the email gateway. The email claimed to be an invoice from a known supplier. The analyst begins analysis in their isolated REMnux / Flare-VM lab environment.

Phase 1 — Initial triage (15 minutes)

Triage: Hash lookup and document analysis

SHA256 hash submitted to VirusTotal: 47 of 73 engines flag it as malicious. Family: Emotet loader → TrickBot. Tagged: macro dropper, banking trojan.

olevba tool extracts the VBA macro from the document. The macro is obfuscated — variable names are random strings. After de-obfuscating, it reveals a PowerShell command that downloads and executes a second-stage payload

Phase 2 — Static analysis of the dropped PE (45 minutes)

Static: PE analysis of the downloaded payload

The downloaded payload is a 64-bit PE file. Detect-It-Easy identifies it as packed with a custom packer — no standard packer signature matches.

PE Studio analysis of the IAT reveals key imports: VirtualAllocEx, WriteProcessMemory, CreateRemoteThread (process injection), WinHttpConnect, WinHttpSendRequest (C2 comms), and CryptEncrypt (data encryption before exfiltration).

Phase 3 — Dynamic analysis (30 minutes in sandbox)

Dynamic: Cuckoo sandbox execution

Running in Cuckoo sandbox: The malware injects itself into svchost.exe using the classic VirtualAllocEx → WriteProcessMemory → CreateRemoteThread triad confirmed in static analysis.

Procmon captures registry persistence: HKCU\Software\Microsoft\Windows\CurrentVersion\Run key written with the malware path. INetSim captures DNS queries to all decoded C2 domains and an HTTP POST to /gate.php — a classic

The report and IOCs produced

IOC Type	Value	Significance
SHA256	47a8c3...f92e1 (document)	Email attachment hash — add to email gateway block list
SHA256	9b3d1f...4a72c (payload)	Dropper hash — add to EDR and AV block list
IP Address	185.220.101.xx, 195.123.xx.xx	C2 server IPs — block at perimeter firewall

IOC Type	Value	Significance
Domain	updates.microsoft-cdn.net (fake)	C2 domain — add to DNS sinkholes and proxy block list
Registry Key	HKCU\...\Run\WindowsDefender_x64	Persistence mechanism — hunt for this key across estate
URL Pattern	POST /gate.php	C2 communication pattern — add to SIEM detection rule
YARA Rule	BankingTrojan_TrickBot_Injector	Custom YARA rule based on IAT patterns — deploy to scanner

Table 10: Indicators of Compromise (IOCs) extracted from the banking trojan analysis

Total analysis time: approximately 90 minutes.
Output: threat report, 6 IOCs shared with SIEM/firewall/EDR teams, 1 YARA rule.
The same IOCs were shared via MISP with partner banks through sector threat sharing.

12. Breaking In — Getting Your First Role

Malware analysis is not a typical first job. Most people enter through the SOC, incident response or digital forensics — and then develop RE skills alongside their main role. The portfolio you build on the side often matters more than your job title.

Build visible evidence of skill

- **Malware analysis write-ups:** Download samples from MalwareBazaar. Analyse them. Write up your findings in a professional report format — IOCs, YARA rule, TTPs mapped to MITRE ATT&CK.; Publish on a blog or GitHub
- **Crackme challenges:** Complete beginner, intermediate and advanced crackmes from crackmes.one. These demonstrate RE skill without involving real malware. Post your solutions with detailed explanations
- **CTF reverse engineering challenges:** CTFtime.org lists competitions with RE categories. Completed RE challenges with detailed write-ups are respected by hiring managers
- **GitHub:** Share your Python scripts for analysis automation, YARA rules you have written, and any Ghidra scripts you have developed
- **Flare-On challenge:** FireEye (Mandiant) runs an annual RE competition — Flare-On. Completing even the early levels demonstrates real reverse engineering ability

Where to Look	Notes
CyberSecurityJobs.com	Filter by 'malware analyst', 'reverse engineer', 'threat researcher'
LinkedIn	Niche field — recruiters actively search LinkedIn. Connect with malware researchers at Mandiant, CrowdStrike, Recorded Future
Mandiant / CrowdStrike / Recorded Future	Major threat intelligence firms hire regularly. Apply directly — they value portfolio over certs
GCHQ / NCSC / defence contractors	Government intelligence roles — strong demand for RE skills, often requiring SC or DV clearance
Antivirus vendors (Sophos, Kaspersky, ESET)	Labs teams analyse thousands of samples per day. Entry-level lab analyst roles exist
Academic / research positions	Universities and research institutes (e.g. Alan Turing Institute) hire for RE and malware research

Table 11: Where to find malware analysis and reverse engineering roles

Interview questions to prepare for

- Walk me through your process when you receive an unknown suspicious file for analysis.
- What is the Import Address Table and what can it tell you about a malware sample?
- Explain the difference between static and dynamic analysis. When would you use each?
- A sample is packed. How do you approach unpacking it?
- What anti-analysis techniques have you encountered and how do you overcome them?
- Write me a YARA rule that would detect a file containing the string 'Bot ID:' and a PE header.

- What is process hollowing and how would you detect it using Procmon or memory analysis?
- How would you analyse a macro-enabled Office document for malicious content?

The best malware analysis interview answer:
"Here is a full analysis report I wrote on a TrickBot sample from MalwareBazaar."
Published analysis work is the most convincing evidence of real capability.

13. References

1. abuse.ch (2024) *MalwareBazaar — Malware Sample Database.* Available at: https://bazaar.abuse.ch [Accessed: 10 April 2026].

2. ANY.RUN (2024) *Interactive Malware Analysis Sandbox.* Available at: https://any.run [Accessed: 10 April 2026].

3. CW Jobs (2024) *Technology Salary Survey 2024.* Available at: https://www.cwjobs.co.uk/salary-checker [Accessed: 11 April 2026].

4. Flare-VM (2024) *Windows Incident Response and Reverse Engineering Distribution.* Mandiant. Available at: https://github.com/mandiant/flare-vm [Accessed: 11 April 2026].

5. GIAC (2024) *GREM — GIAC Reverse Engineering Malware Certification.* Available at: https://www.giac.org/certifications/reverse-engineering-malware-grem/ [Accessed: 11 April 2026].

6. Ligh, M.H., Case, A., Levy, J. and Walters, A. (2014) *The Art of Memory Forensics.* Indianapolis: Wiley.

7. Malware Unicorn (2024) *Free Malware Analysis Workshops.* Available at: https://malwareunicorn.org [Accessed: 12 April 2026].

8. MITRE (2024) *ATT&CK; Framework — Adversarial Tactics, Techniques and Common Knowledge.* Available at: https://attack.mitre.org [Accessed: 12 April 2026].

9. National Security Agency (2024) *Ghidra — Open Source Reverse Engineering Tool.* Available at: https://ghidra-sre.org [Accessed: 12 April 2026].

10. OpenSecurityTraining2 (2024) *Architecture 1001: x86-64 Assembly.* Available at: https://ost2.fyi [Accessed: 13 April 2026].

11. Russinovich, M., Solomon, D. and Ionescu, A. (2017) *Windows Internals, Part 1.* 7th edn. Microsoft Press.

12. TCM Security (2024) *Practical Malware Analysis and Triage (PMAT).* Available at: https://academy.tcm-sec.com/p/practical-malware-analysis-triage [Accessed: 13 April 2026].

13. VirusTotal (2024) *Online Malware Scanner and Threat Intelligence Platform.* Available at: https://www.virustotal.com [Accessed: 14 April 2026].

Document prepared by **Babashaheer**. Version 1.0 — April 2026. Cybersecurity Career Series — Document 04 of 09.